UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/961,048	09/20/2001	Michal J. Cierniak	42390P11912	5003

8791 7590 06/13/2005

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
LAO, SUE X	

ART UNIT	PAPER NUMBER
2194	

DATE MAILED: 06/13/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/961,048	CIERNIAK, MICHAL J.	
	Examiner	Art Unit	
	Sue Lao	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 February 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *20 September 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *2/22/2005*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 are pending. This action is in response to the amendment filed 2/20/2005. Applicant has amended claims 1, 2, 16 and added claims 19, 20.

2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claim 1-10, 14, 15 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The language of independent claims 1, 9, 14 raises a question as to whether the claim is directed merely to an abstract idea that is not tied to a technological art, environment or machine which would result in a practical application producing a useful, concrete and tangible result to form the basis of statutory subject matter under 35 U.S.C. 101.

Independent claims 1, 9, 14 do not appear to require any computer hardware to implement the claimed invention. These claims appear to define the metes and bounds of an invention comprised of software alone. There is no support (i.e., explicitly claimed computer hardware) in the body of the claims. Software alone, without a machine, is incapable of transforming any physical subject matter by chemical, electrical, or mechanical acts. If the "acts" of a claimed process manipulate only numbers, abstract concepts or ideas, or signals representing any of the foregoing, the acts are not being applied to appropriate subject matter. In re Schrader, 22 F.3d 290 at 294-95, 30 USPQ2d 1455 at 1458-59 (Fed. Cir. 1994). Transformation of data by a machine constitutes statutory subject matter if the claimed invention as a whole accomplishes a practical application. That is, it must produce a "useful, concrete and tangible result." State Street, 149 F.3d 1368, 1373, 47 USPQ2d 1596 at 1600-02 (Fed. Cir. 1998).

MPEP 2106. State Street required transformation of data by a machine before it applied the "useful, concrete, and tangible test." However, State Street does not hold that a "useful, concrete and tangible result" alone, without a machine, is sufficient for statutory subject matter. State Street, 149 F.3d at 1373, 47 USPQ2d at 1601.

Claims 1-10, 14, 15 are rejected under 35 U.S.C. 101 because the claimed invention, appearing to be comprised of software alone without claiming associated computer hardware required for execution, is not supported by either a specific and substantial asserted utility (i.e., transformation of data) or a well established utility (i.e., a practical application).

5. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 1-10, 14, 15 are also rejected under 35 U.S.C. 112, first paragraph. Specifically, since the claimed invention is not supported by either a specific and substantial asserted utility or a well established utility for the reasons set forth above, one skilled in the art clearly would not know how to use the claimed invention.

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-10, 14, 15 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential elements, such omission amounting to a gap between the elements. See MPEP § 2172.01. The omitted elements are computer hardware necessary to execute the claimed software and render the invention operative.

7. Claims 1, 4-18, 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stoodley et al (U S Pat. 6,182,282).

As to claim 1, Stoodley teaches (hybrid VFT implementation) a data structure comprising: a table (hybrid VFT structure, fig. 2A, 4) for virtual method dispatch (virtual function table) and type identification (classes B, C, D, fig. 4), wherein the table includes a plurality of pointers (addresses of B::X(), C::Y()), wherein the plurality of pointers point to a plurality of classes (class B, class C) and wherein the plurality of classes include at least one unified type hierarchy (B is of type 'old", C is of type "new"). See col. 8, line 66 – col. 9, line 40; fig.s 2A, 4, 5. Stoodley further teaches a plurality of names (fig. 4) and a plurality of programming languages (C++, Java and CFRONT, col. 1, lines 14-15; col. 3, lines 23-27).

As to the plurality of names being from a plurality of programming languages for one implementation, Stoodley teaches the hybrid VFT structure supports classes compiled with compilers using different virtual function table layouts and/or different function member call protocols (abstract, col. 11, line 60 col. 12, line 6) through the use of thunk / pointer adjustment (col. 3, lines 40-57). It is noted that a language's specification defines its virtual function table layout and its function member call protocol. Stoodley teaches a plurality of programming languages, such as C++, Java and CFRONT, which, to one of ordinary skill in the art, support virtual dispatching but have different virtual function table layouts and/or different function member call protocols. Therefore, it would have been obvious to implement the classes of the hybrid VFT structure with such programming languages, ie, first programming language and a second programming language. When the teaching is modified as such, the names would have been from a plurality of programming languages.

As to claim 4, Stoodley teaches the unified type hierarchy (classes B, C, D, fig. 1) contains classes of different/incompatible virtual function data structures (col. 1, lines 8-10; produced by old and new compilers, col. 8, line 66 – col. 9, line 1). Stoodley also teaches that such different/incompatible virtual function data structures result from multiple object-oriented programming environments/languages including C++, Java

and CFRONT (col. 1, lines 14-15; col. 3, lines 23-27). Therefore, it would have been obvious to implement such different/incompatible virtual function data structures with first programming language and a second programming language. When the teaching is modified as such, the unified type hierarchy would have included a data structure (hybrid VFT) recognizable by a first programming language and a second programming language.

As to claims 5-6, the teaching of Stoodley as modified (note discussion of claim 4) would have been applicable to / use for two or more hierarchical programming languages, which are object-oriented programming languages.

As to claim 7, note discussion of claims 4-6, and Stoodley teaches the object-oriented programming languages include Java, C++. Since C#, Smalltalk and Eiffel are common object-oriented programming languages with differing virtual function data structures which is the subject of Stoodley, it would have been obvious to include these languages into the system of Stoodley as modified.

As to claim 8, Stoodley teaches a root (flag) identifying each programming environment (new or old, col. 11, lines 45-48). Note discussion of claim 4 for implementing differing programming environments with differing programming languages.

As to claim 9, Stoodley teaches a method of identifying equivalent data structures (virtual functions) comprising: receiving a plurality of data structures (virtual functions X(), Y(), Z() of classes B, C, D), comparing the implementation of each one of the plurality of data structures (determine inherited virtual functions) [also inherent to the identifying two of the plurality of data structures having identical implementation]; and identifying at least two of the plurality of data structures that have identical implementations (X() of class D and X() of class B; Y() of class D and Y() of class C). See Col. 8, lines 16-52. Regarding that each one of the plurality of data structures are from a different one of a plurality of programming languages, note discuss of claim 4 for implementing each of the differing/incompatible virtual function data structures with a respective programming language.

As to claims 10, 13, 17, note discussion of claim 7.

As to claim 11, it is covered by claim 9 except for processor, I/O system, bus and memory, which would have been inherent to the system of Stoodley, or obvious to be included.

As to claim 12, Stoodley teaches execution in Java environment (col. 1, lines 14-16), which is typically a distributed architecture. Therefore, it would have been obvious to include a network adapter into the system of Stoodley as modified.

As to claim 14, note discussion of claim 9 for steps of receiving, comparing and identifying. Stoodley further teaches creating a unified data structure (hybrid VFT) wherein the unified data structure includes: a single implementation (function X()) of the identified at least two data structures (X() of class B and X() of class D); and a plurality of names of the identified at least two data structures (B::X() and D::X()). See Col. 8, lines 16-52; col. 8, line 66 – col. 9, line 40; fig.s 2A, 4, 5.

As to claims 15, 18, 20, note discussion of claim 1 for names and languages. In particular, Stoodley teaches a plurality of names (B::X() and D::X(), fig. 4) and a plurality of programming languages (C++, Java and CFRONT, col. 1, lines 14-15; col. 3, lines 23-27). As to the names corresponding to a plurality of programming languages, Stoodley teaches the hybrid VFT structure supports classes compiled with compilers using different virtual function table layouts and/or different function member call protocols (abstract, col. 11, line 60 col. 12, line 6) through the use of thunk / pointer adjustment (col. 3, lines 40-57). It is noted that a language's specification defines its virtual function table layout and its function member call protocol. Stoodley teaches a plurality of programming languages, such as C++, Java and CFRONT, which, to one of ordinary skill in the art, support virtual dispatching but have different virtual function table layouts and/or different function member call protocols. Therefore, it would have been obvious to implement the classes of the hybrid VFT structure with such programming languages, ie, first programming language and a second programming language. When the teaching is modified as such, the names would have corresponded to a plurality of programming languages.

As to claim 16, note discussion of claim 14. Further note discussion of claim 11 for processor, I/O system, bus and memory.

8. Claims 2, 3, 19 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 101 and 112 set forth in this Office action and to include all of the limitations of the respective base claims and any intervening claims, subjected to a final search.

9. Applicant's arguments filed 2/22/2005 have been fully considered but they are not persuasive.

As to the argument regarding a plurality of names from a plurality of programming languages for one implementation (remarks, page 8), Stoodley teaches a plurality of names (B::X() and D::X(), fig. 4) and teaches a plurality of programming languages (C++, Java and CFRONT, col. 1, lines 14-15; col. 3, lines 23-27). As to the plurality of names being from a plurality of programming languages for one implementation, Stoodley teaches the hybrid VFT structure supports classes compiled with compilers using different virtual function table layouts and/or different function member call protocols (abstract, col. 11, line 60 col. 12, line 6) through the use of thunk / pointer adjustment (col. 3, lines 40-57). It is noted that a language's specification defines its virtual function table layout and its function member call protocol. Stoodley teaches a plurality of programming languages, such as C++, Java and CFRONT, which, to one of ordinary skill in the art, support virtual dispatching but have different virtual function table layouts and/or different function member call protocols. Therefore, it would be obvious to implement the classes of the hybrid VFT structure with such programming languages, ie, by a first programming language and a second programming language. When the teaching is modified as such, the names would be from / correspond to a plurality of programming languages.

Regarding the argued comparing (remarks, page 9), it would be inherent to / obvious to the step of identifying two data structures having identical implementations (X() of class D and X() of class B; Y() of class D and Y() of class C). See Col. 8, lines 16-52. It is noted that equality is typically determined by comparison operations such as IsEqual() which compares corresponding members of two objects/classes.

As to the argued recognizable by multiple languages (remarks, page 10), it is the teaching of Stoodley as applied to claim 1, that meets this limitations. As discussed for claim 1, Stoodley teaches a plurality of names (B::X() and D::X(), fig. 4) and teaches a plurality of programming languages (C++, Java and CFRONT, col. 1, lines 14-15; col. 3, lines 23-27). As to the plurality of names being from a plurality of programming languages for one implementation, Stoodley teaches the hybrid VFT structure supports classes compiled with compilers using different virtual function table layouts and/or different function member call protocols (abstract, col. 11, line 60 col. 12, line 6) through the use of thunk / pointer adjustment (col. 3, lines 40-57). It is noted that a language's specification defines its virtual function table layout and its function member call protocol. Stoodley teaches a plurality of programming languages, such as C++, Java and CFRONT, which, to one of ordinary skill in the art, support virtual dispatching but have different virtual function table layouts and/or different function member call protocols. Therefore, it would be obvious to implement the classes of the hybrid VFT structure with such programming languages, ie, by a first programming language and a second programming language. When the teaching is modified as such, the names would be from / correspond to a plurality of programming languages. In other words, as modified, the hybrid VFT structure would support / recognizable by more than one programming languages.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sue Lao whose telephone number is (571) 272-3764. A voice mail service is also available at this number. The examiner's supervisor, SPE Meng-Ai An, can be reached on (571) 272 3756. The examiner can normally be reached on Monday - Friday, from 9AM to 5PM. The fax phone number for the organization where this application or proceeding is assigned is (703) 872 9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

June 6, 2005

SUE LAO
PRIMARY EXAMINER